UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/709,705	05/24/2004	Jonathan K. Weedon	BOR-011	3704

64107 7590 05/07/2009

KOKKA & BACKUS, PC
200 PAGE MILL ROAD
SUITE 103
PALO ALTO, CA 94306

EXAMINER	
WANG, RONGFA PHILIP	
ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
05/07/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/709,705	WEEDON ET AL.	
	Examiner	Art Unit	
	PHILIP WANG	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *06 March 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-13,15,17-34,36 and 38-47* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-13,15,17-34,36 and 38-47* is/are rejected.

7)☒ Claim(s) *15, 17, 36 and 38* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Detail Action

1. This office action is in response to amendment filed on 3/6/2009.

2. Per Applicant's request, claims 1, 2, 22, 31, 44 and 45 are amended; claims 14, 16, 35, and 37 are canceled.

3. The objection to specification has been withdrawn in view of the Applicant's amendment to Specification.

4. The 35 USC §112 rejections of claims 2, 44, and 45 are withdrawn in view of the Applicant's amendment to the claims.

5. Claims 1-13, 15, 17-34, 36, 38-47 are pending.

Claim Objections

6. Claims 15, 17, 36 and 38 are objected to because of the following informalities: claims 15 and 17 depend on claim 14 which has been canceled. Claims 36 and 38 depend on claim 35 which has been canceled. Appropriate correction is required.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1, 3-5, 7-13, 15 17-22, 24-27, 29-34, 36, 38-43, 46 and 47 are rejected under 35 U.S.C. 102(b) as being unpatentable by Kanamori (US Patent No. 6,167,565) in view of Moore et al. (herein Moore, USPTN 6,408,342).

As per claim 1,

Kanamori discloses

A system for translation of data types between a first application in a first language and a second application in a second language, the system comprising (c3:64-c4:49):

- A computer having at least one processor and a memory (c1:6-8, "…a computer system…");
- a formal mapping between data types of the first language and data types of the second language (for example, c4: 3-6, "…a mapping from a data type in one programming language to a corresponding data type in another programming language…");
- translators for translating data types between the first language and the second language based on the formal mapping; a translation mapping to the translators based on actual data types of the first application and formal data types of the second application (c4:6-9, "This mapping specifies custom marshaling code…that can be used for converting…data type to …the corresponding data type…""); and
- a module for automatically selecting an appropriate translator for translating between a particular data type in the first language and a data type in the second language based on the translation mapping in response to invocation of a method of the first application with the particular data type (c3: 65-c4:2,

"…customer marshaling of parameters during invocations…"; for example, c4:17-20, "…the translator invokes the custom marshaling code to convert…" where the conversion of data types is by executing code, and therefore considered automatically.).

Kanamori does not specifically discloses

- wherein the module for selecting the appropriate translator is configured to perform at least a two-level lookup in the translation mapping to select the appropriate translator, wherein at least one level of the two-level lookup includes a lookup of an inheritance hierarchy of the actual type to select the appropriate translator.

However, Moore discloses

- wherein the module for selecting the appropriate translator is configured to perform at least a two-level lookup in the translation mapping to select the appropriate translator, wherein at least one level of the two-level lookup includes a lookup of an inheritance hierarchy of the actual type to select the appropriate translator(c91, 92: claims 8 &9 -- 8. The communication framework of claim 3, further comprising: an OutStream class for declaring an interface including at least one primitive marshaler and a composite data type marshaler, wherein each remote procedure call transport defines an implementation of the interface of the OutStream class for marshaling arguments associated with the performable operation. 9. The communication framework of claim 8, further comprising: a composite data type base class and at least one transport independent marshaler, wherein the implementation of the interface of the OutStream class

recognizes a composite data type object deriving from the composite data type base class, and wherein each remote procedure call transport invokes selectively the at least one transport independent marshaler to marshal any composite data type object; where a second lookup employing inheritance hierarchy is discloses, where data type recognized by the composite data type base class will invoke the transport independent marshaler.)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Moore into the teachings of Kanamori to include the limitation disclosed by Moore. The modification would be obvious to one of ordinary skill in the art to want to provide a translator associated with a base class so all derived data type of the base class can invoke the associated translator. The associated translator can be reused for all data type derived from the base class.

As per claim 3,

the rejection of claim 1 is incorporated,

Kanamori discloses

- wherein the formal mapping comprises a mapping between formal types of the first language and formal types of the second language (c4: 6-9, "...mapping from a data type...to a corresponding data type...").

As per claim 4,

the rejection of claim 3 is incorporated,

Kanamori discloses

- wherein the formal types comprise static types (c10: 26, where a static type is shown.).

As per claim 5,

the rejection of claim 1 is incorporated,

Kanamori discloses

- wherein the formal mapping comprises a many-to-one mapping (for example c4, Table 1, first and third rows where both [in]COMTYPE and [in]COMTYPE* map to JAVATYPE).

As per claim 7,

the rejection of claim 1 is incorporated,

Kanamori discloses

- wherein the translators read data of a first type and write data of a second type (c4:64-c4:49).

As per claim 8,

the rejection of claim 1 is incorporated,

Kanamori discloses

- wherein the translators include a mechanism for determining the actual type in the first language that a particular translator supports (c4: 45-46, "…determines whether customer marshaling has been defined for any of the parameters...").

As per claim 9,

the rejection of claim 1 is incorporated,

Kanamori discloses

- wherein the translators include a mechanism for determining the formal type in the second language that a particular translator supports (see Table 1).

As per claim 10,

the rejection of claim 1 is incorporated,

Kanamori discloses

- wherein the translators provide information needed for creating the translation mapping (c4:3-10, "The custom marshaling system…provide to the translator a mapping...").

As per claim 11,

the rejection of claim 1 is incorporated,

Kanamori discloses

- wherein the translators translate return values received from the second application into a format appropriate for the first application (c4:17-20, "…Upon returning from the function,…convert the formal parameter back into the data type of the actual parameter.").

As per claim 12,

the rejection of claim 1 is incorporated,

Kanamori discloses

- wherein the translation mapping provides for navigation from an object of the first application to a formal type of the second application's environment(see Table 1 for example).

As per claim 13,

the rejection of claim 1 is incorporated,

Kanamori discloses

- wherein the translation mapping comprises a mapping from actual type of the first application and formal type of the second application to a particular translator(c4: 6-9, "This mapping specifies custom marshaling code...that can be used for converting parameter of the data type to a parameter of the corresponding data type, and vice versa.").

As per claim 15,

the rejection of claim 14 is incorporated,

Kanamori discloses

- wherein the two level lookup includes a first level lookup based on actual data type of the first application(see Table 1).

As per claim 17,

the rejection of claim 14 is incorporated,

Kanamori discloses

- wherein the two level lookup includes a second level lookup based on formal data type of the second application (c4: 6-16, "This mapping specifies custom marshaling code...that can be used for converting parameter of the data type to a parameter of the corresponding data type, and vice versa…the translator retrieves the specified custom marshaling code."; see Table 1 shows two levels: first function name, "Foo", then data type).

As per claim 18,

the rejection of claim 17 is incorporated,

Kanamori discloses

- wherein the second level lookup selects the appropriate translator from a set of translators determined by the first level lookup (c4: 6-16, "This mapping specifies custom marshaling code...that can be used for converting parameter of the data type to a parameter of the corresponding data type, and vice versa…the translator retrieves the specified custom marshaling code."; see Table 1 shows two levels: first function name, "Foo", then data type).

As per claim 19,

the rejection of claim 1 is incorporated,

Kanamori discloses

- wherein the module for selecting an appropriate translator determines if the mapping includes at least one translator for the particular data type (c4: 6-16, "This mapping specifies custom marshaling code...that can be used for converting parameter of the data type to a parameter of the corresponding data type, and vice versa…the translator retrieves the specified custom marshaling code.";).

As per claim 20,

the rejection of claim 1 is incorporated,

Kanamori discloses

- wherein the module for selecting an appropriate translator determines if the mapping includes at least one translator for interfaces of the particular data type (c4: 6-16, "This mapping specifies custom marshaling code...that can be used for converting parameter of the data type to a parameter of the corresponding data type, and vice versa…the translator retrieves the specified custom marshaling code.";).

As per claim 21,

the rejection of claim 1 is incorporated,

Kanamori discloses

- wherein the module for selecting an appropriate translator determines if the mapping includes at least one translator for base types of the particular data

type (c4: 6-16, "This mapping specifies custom marshaling code...that can be used for converting parameter of the data type to a parameter of the corresponding data type, and vice versa…the translator retrieves the specified custom marshaling code.";).

As per claim 22,

Kanamori discloses

A method for translation of data types between a first component in a first language and a second component in a second language, the method comprising (c3:64-c4:49):

- defining a formal mapping between data types of the first language and data types of the second language (for example, c4: 3-6, "…a mapping from a data type in one programming language to a corresponding data type in another programming language…");
- implementing translators based on the formal mapping for translating data types between the first language and the second language (c4:6-9, "This mapping specifies custom marshaling code…that can be used for converting…data type to …the corresponding data type…"");
- producing a programming interface for the first component based upon the formal mapping and the second component's programming interface; generating a translation mapping to the translators based on actual data types of the first component and formal data types of the second component as defined in the

first component's programming interface (c4:25-49, where example based on C++ COM and Java VM is disclosed);

- in response to invocation of a method defined in the first component's programming interface with a particular data type, automatically selecting a translator based on the translation mapping and the particular data type (c3: 65-c4:2, "…customer marshaling of parameters during invocations…"); and translating the particular data type to a data type of the second language using the selected translator (c4: 12- 24, where customer marshaling code is used for converting data types is disclosed; where the conversion of data types is by executing code, and therefore considered automatically.).

Kanamori does not specifically discloses

- wherein the module for selecting the appropriate translator is configured to perform at least a two-level lookup in the translation mapping to select the appropriate translator, wherein at least one level of the two-level lookup includes a lookup of an inheritance hierarchy of the actual type to select the appropriate translator.

However, Moore discloses

- wherein the module for selecting the appropriate translator is configured to perform at least a two-level lookup in the translation mapping to select the appropriate translator, wherein at least one level of the two-level lookup includes a lookup of an inheritance hierarchy of the actual type to select the appropriate translator(c91, 92: claims 8 &9 -- 8. The communication framework of claim 3, further comprising: an OutStream class for declaring an

interface including at least one primitive marshaler and a composite data type marshaler, wherein each remote procedure call transport defines an implementation of the interface of the OutStream class for marshaling arguments associated with the performable operation. 9. The communication framework of claim 8, further comprising: a composite data type base class and at least one transport independent marshaler, wherein the implementation of the interface of the OutStream class recognizes a composite data type object deriving from the composite data type base class, and wherein each remote procedure call transport invokes selectively the at least one transport independent marshaler to marshal any composite data type object; where a second lookup employing inheritance hierarchy is discloses, where data type recognized by the composite data type base class will invoke the transport independent marshaler.)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Moore into the teachings of Kanamori to include the limitation disclosed by Moore. The modification would be obvious to one of ordinary skill in the art to want to provide a translator associated with a base class so all derived data type of the base class can invoke the associated translator. The associated translator can be reused for all data type derived from the base class.

As per claim 24,

the rejection of claim 22 is incorporated;

Kanamori discloses

- wherein the first component comprises a first component of an application and the second component comprises a second component of the application(c4:25-65).

As per claim 25,

the rejection of claim 22 is incorporated;

Kanamori discloses

- wherein the first component and the second component operate within a single process(see FIG. 1).

As per claim 26,

the rejection of claim 22 is incorporated;

Kanamori discloses

- wherein the defining step includes defining a mapping between formal types of the first language and formal types of the second language(c4: 2-9).

As per claim 27,

the rejection of claim 22 is incorporated;

Kanamori discloses

- wherein the defining step includes defining a many-to-one mapping (for example c4, Table 1, first and third rows where both [in]COMTYPE and [in]COMTYPE* map to JAVATYPE).

As per claim 29,

the rejection of claim 22 is incorporated;

Kanamori discloses

- wherein the implementing step includes implementing a translator reading data of a first type and writing data of a second type(c4: 2-20).

As per claim 30,

the rejection of claim 22 is incorporated;

Kanamori discloses

- wherein the implementing step includes indicating the actual type in the first language that a particular translator supports(c4:2-20, also see Table 1).

As per claim 31,

the rejection of claim 22 is incorporated;

Kanamori discloses

- wherein the implementing step includes indicating the formal type in the second language that a particular translator supports(c4:2-20, also see Table 1).

As per claim 32,

the rejection of claim 22 is incorporated;

Kanamori discloses

- wherein the generating step includes generating the translation mapping based, at least in part, on information provided by the translators(c4:2-20).

As per claim 33,

the rejection of claim 22 is incorporated;

Kanamori discloses

- wherein the translation mapping provides for navigation from an object of the
- first component to the formal type of the second component's environment (see Table 1 for example).

As per claim 34,

the rejection of claim 22 is incorporated;

Kanamori discloses

- wherein the translation mapping comprises a mapping from actual type of the first component and formal type of the second component to a particular translator(c4:2-64).

.

As per claim 36,

the rejection of claim 35 is incorporated;

Kanamori discloses

- wherein the two level lookup includes a first level lookup based on actual data type of the first component (c4: 6-16, "This mapping specifies custom marshaling code...that can be used for converting parameter of the data type to a parameter of the

corresponding data type, and vice versa…the translator retrieves the specified custom marshaling code..";).

.

.

As per claim 38,

the rejection of claim 35 is incorporated;

Kanamori discloses

- wherein the two level lookup includes a second level lookup based on formal data type of the second component (c4: 6-16, "This mapping specifies custom marshaling code...that can be used for converting parameter of the data type to a parameter of the corresponding data type, and vice versa…the translator retrieves the specified custom marshaling code.."; see Table 1 shows two levels: first function name, "Foo", then data type).

.

As per claim 39,

the rejection of claim 38 is incorporated;

Kanamori discloses

- wherein the second level lookup includes selecting a translator from a set of translators determined by the first level lookup based on formal data type (c4: 6-16, "This mapping specifies custom marshaling code...that

can be used for converting parameter of the data type to a parameter of the corresponding data type, and vice versa…the translator retrieves the specified custom marshaling code.."; see Table 1 shows two levels: first function name, "Foo", then data type).

As per claim 40,

the rejection of claim 22 is incorporated;

Kanamori discloses

- wherein the selecting step includes determining if the translation mapping includes at least one translator for the particular data type(c4:1-20).

As per claim 41,

the rejection of claim 22 is incorporated;

Kanamori discloses

- wherein the selecting step includes determining if the translation mapping includes at least one translator for interfaces of the particular data type(c4:2-64).

As per claim 42,

the rejection of claim 22 is incorporated;

Kanamori discloses

- the selecting step includes determining if the translation mapping includes at least one translator for base types of the particular data type(c4:64).

As per claim 43,

The rejection of claim 22 is incorporated;

Kanamori discloses

- translating return values received from the second component
- into a data type of the first component's environment using the selected translator (c4:17-20, "…Upon returning from the function,…convert the formal parameter back into the data type of the actual parameter.").

As per claim 46, it claims similar limitation as method claim 22 and is rejected for similar reason set forth in connection of the rejection of claim 22 above.

As per claim 47, it claims similar limitation as method claim 22 and is rejected for similar same reason set forth in connection of the rejection of claim 22 above.

8. Claims 2, 44 and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kanamori (US Patent No. 6,167,565) in view of Moore et al. (herein Moore, USPTN 6,408,342) and further in view of Vargas (US PGPub. No. 2004/0103405).

As per claim 2,

the rejection of claim 1 is incorporated,

- the second language comprises a language that is configured to conform with a JAVA language specification(c2 46-47, “Java programming language…”) .

Kanamori/Moore does not specifically disclose

- the first language comprises C#.

However, Vargas discloses

- the first language comprises C#([0006], “…C#...”).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Vargas into the teachings of Kanamori/Moore to include the limitation discloses by Vargas . The modification would be obvious to one of ordinary skill in the art to want to minimize the differences between programming languages as suggested by Vargas ([0013]).

As per claim 44,

the rejection of claim 22 is incorporated,

- the second language is a language that is configured to conform with a JAVA language specification (c2 46-47, “Java programming language…”) .

Kanamori does not specifically disclose

- the first language comprises C#.

However, Vargas discloses

- the first language comprises C#([0006], “…C#...”).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Vargas into the teachings of Kanamori/Moore to include the limitation discloses by Vargas . The modification would be

obvious to one of ordinary skill in the art to want to minimize the differences between programming languages as suggested by Vargas ([0013]).

As per claim 45,

the rejection of claim 22 is incorporated,

- the first language is a language other that is configured to conform with a JAVA language specification (c2 46-47, "Java programming language…") .

Kanamori does not specifically disclose

- the second language comprises C#.

However, Vargas discloses

- the second language comprises C#([0006], "…C#...").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Vargas into the teachings of Kanamori to include the limitation discloses by Vargas . The modification would be obvious to one of ordinary skill in the art to want to minimize the differences between programming languages as suggested by Vargas ([0013]).

9. Claims 6, 23, 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kanamori (US Patent No. 6,167,565) in view of Moore et al. (herein Moore, USPTN 6,408,342) and further in view of Beisiegel et al. (US PGPub. No. 2004/0177360).

As per claim 6,

the rejection of claim 1 is incorporated,

Kanamori/Moore does not specifically disclose

- wherein the translators marshal translated data into a wire format for transfer from the first application to the second application across a network.

However, Beisiegel et al. disclose

- wherein the translators marshal translated data into a wire format for transfer from the first application to the second application across a network ([0012], "…conversion to and from an arbitrary native wire data format…";[0034], "…a networked computing device, is in communication with other networked computing device.).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Beisiegel et al. into the teachings of Kanamori/Moore to include the limitation discloses by Beisiegel et al. . The modification would be obvious to one of ordinary skill in the art to want to enable cooperating components of an enterprise as suggested by Beisiegel et al. ([0005]).

As per claim 23,

the rejection of claim 22 is incorporated;

Kanamori/Moore does not specifically disclose

- wherein the first component comprises an application on a first machine and the second component comprises an application on a second machine.

However, Beisiegel et al. disclose

- wherein the first component comprises an application on a first machine and the second component comprises an application on a second machine([0034], "…a networked computing device, is in communication with other networked computing device.).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Beisiegel et al. into the teachings of Kanamori/Moore to include the limitation discloses by Beisiegel et al. . The modification would be obvious to one of ordinary skill in the art to want to enable cooperating components of an enterprise as suggested by Beisiegel et al. ([0005]).

As per claim 28,

the rejection of claim 22 is incorporated;

Kanamori/Moore does not specifically disclose

- wherein the implementing step includes implementing a translator for marshaling translated data into a wire format for transfer from the first component to the second component across a network.

However, Beisiegel et al. disclose

- wherein the implementing step includes implementing a translator for marshaling translated data into a wire format for transfer from the first component to the second component across a network([0012], "…conversion to and from an arbitrary native wire data format…";[0034], "…a

networked computing device, is in communication with other networked computing device.).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the teachings of Beisiegel et al. into the teachings of Kanamori/Moore to include the limitation discloses by Beisiegel et al. . The modification would be obvious to one of ordinary skill in the art to want to enable cooperating components of an enterprise as suggested by Beisiegel et al. ([0005]).

Response to Amendment

10. Applicant's arguments with respect to claims have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

It is noted that any citation [[s]] to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. [[See, MPEP 2123]]

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Philip Wang whose telephone number is 571-272-5934. The examiner

can normally be reached on Mon - Fri 8:00AM - 4:00PM. Any inquiry of general nature or relating to the status of this application should be directed to the TC2100 Group receptionist: 571-272-2100.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Philip R. Wang/ 5/7/2009